

## HW\_3. Измерение производительности различных имплементаций счетчиков

Измерение проводились на машине со следующими характеристиками:

### Просмотр основных сведений о вашем компьютере

#### Издание Windows

Windows 7 Профессиональная

© Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

Service Pack 1

[Получить доступ к дополнительным функциям, установив новый выпуск Windows 7](#)

#### Система

Оценка:

5,9

Индекс производительности Windows

Процессор:

Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz 2.39 GHz

Установленная память (ОЗУ):

8,00 ГБ (7,90 ГБ доступно)

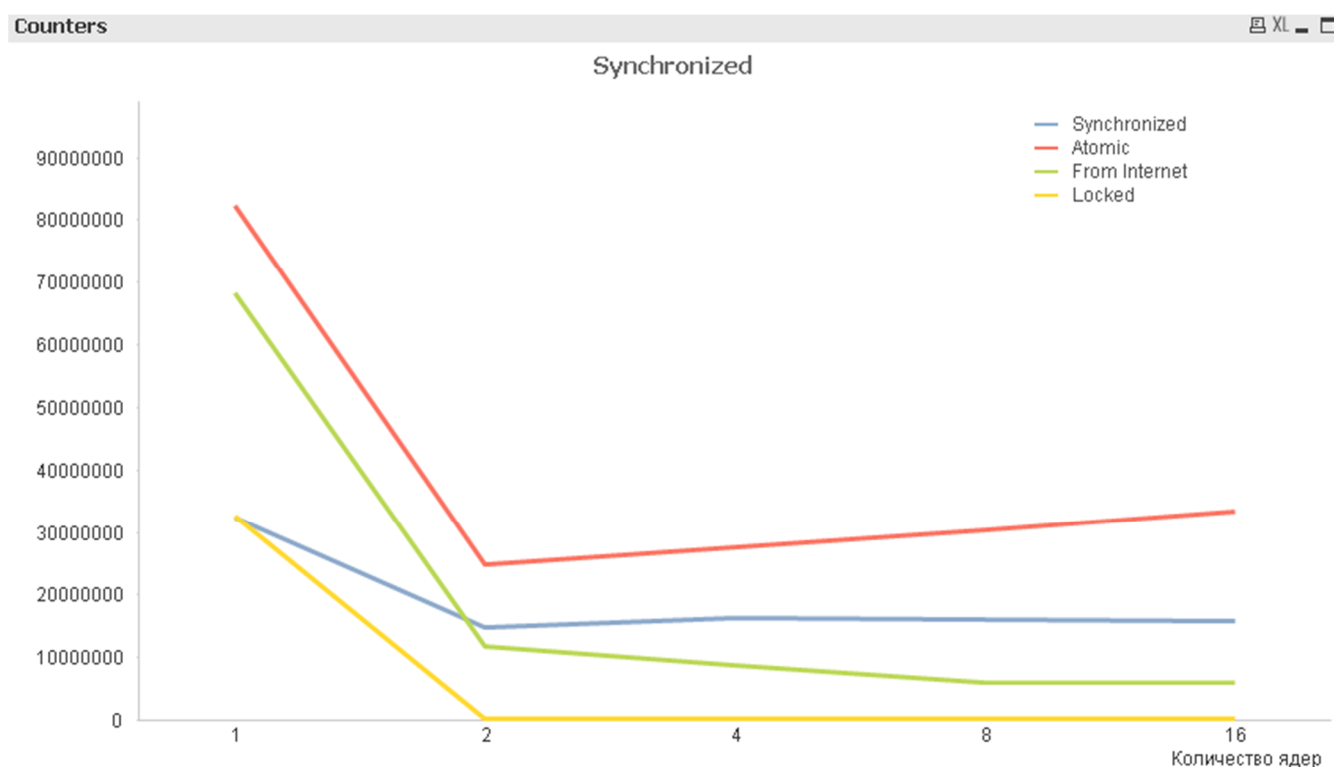
Тип системы:

64-разрядная операционная система

(4 ядра, 8 потоков)

Ссылка на репозиторий с кодом: <https://github.com/EnjeruSan/Counter>

Результат измерения эффективности:



Из графиков можно сделать вывод об эффективности AtomicInteger, которая достигается за счет использования операции incrementAndGet. Она оптимизируется за счет использования *compare-and-swap, CAS*), атомарную инструкцию, которую поддерживает большинство современных процессоров. Эти инструкции работают гораздо быстрее, чем синхронизация с помощью блокировок.

Счетчик "from Internet" тоже использует AtomicInteger, но проигрывает в эффективности за счет дополнительных сравнений.

Блокирование каких-либо секций намного менее эффективно в данном случае, хотя synchronized не так плох, видимо, за счет оптимизаций.

Производительность счетчиков ожидаемо падает с ростом количества потоков, пытающихся использовать его, так как всё же счётчик - это критическая секция, и эффективность программы не может улучшаться при увеличении количества потоков, пытающихся пролезть в "бутылочное горлышко"